APPLICATION

FOR

UNITED STATES LETTERS PATENT

APPLICANT NAME R. V. PATIL

TITLE SYSTEM AND METHOD FOR DYNAMICALLY PRESENTING ACTIONS APPROPRIATE TO A SELECTED DOCUMENT IN A VIEW

DOCKET NO LOT920010028US1

INTERNATIONAL BUSINESS MACHINES CORPORATION

CERTIFICATE OF MAILING UNDER 37 CFR 1.10

I hereby certify that, on the date shown below, this correspondence is being deposited with the United States Postal Service in an envelope addressed to the Box Patent Application, U. S. Patent and Trademark Office, P.O. Box 2327, Arlington, VA 22202 as "Express Mail Post Office to Addressee" on 9 Jan 2002
Mailing Label No. *****EJ410176524 US Name of person mailing paper: Judith A. Beckstrand

Signature Judith A Beckstrand Date 9 Jan 2002

SYSTEM AND METHOD FOR DYNAMICALLY PRESENTING ACTIONS APPROPRIATE TO A SELECTED DOCUMENT IN A VIEW

Background of the Invention

Technical Field of the Invention

This invention pertains to graphical user interfaces. More particularly, it relates to dynamically generating actions which can be performed on a selected document in a view.

Background Art

An action bar is a bar across the top of a Notes view or document that allows a user to execute agents, that is, sets of instructions, by clicking a button. An action is an agent that is specifically designed to be included in an action bar. In Lotus Notes, using an action builder dialog box, an interactive window may be designed to facilitate user control. One or more actions can be combined into an agent.

In form, view and folder design, agents support formula, simple action, LotusScript and JavaScript.

Previously, no matter which document is selected in a view, the actions which are displayed at the top of the view are the same. Some actions may be greyed out, but no matter which document is selected the same actions are greyed out as not selectable. Designers have not been able to display actions which apply to a particular document when that document is selected.

It is an object of the invention to provide an improved system and method for presenting actions which may be executed with respect to a document in a view.

It is an object of the invention to provide a system and method for dynamically evaluating actions which may be executed with respect to successive documents in a view.

It is an object of the invention to provide a system and method whereby actions which may be executed with respect to a view may be dynamically evaluated and selectively hidden or shown for successive view panel content.

Summary of the Invention

A system and method for presenting actions appropriate to a selected document. During action creation, a hide-when formula is determined for each dynamic action. During view creation, a plurality of actions applicable to the view are defined and, responsive to at least one action being a dynamic action, the view is characterized as to be evaluated. Responsive to user selection of a next document for presentation in the view, which next document is characterized as to be evaluated, actions appropriate to the view for this next document are presented.

Other features and advantages of this invention will become apparent from the following detailed description of the presently preferred embodiment of the invention, taken in conjunction with the accompanying drawings.

Brief Description of the Drawings

Figure 1 is a schematic representation of a workstation user interface.

Figures 2 and 3 are schematic representations of views arranged according to a preferred embodiment of the invention for dynamically evaluating action list content for successive documents in a view.

Figure 4 is a flow chart illustrating method steps for creation of static and dynamic actions.

Figure 5 is a schematic representation of an action record.

Figure 6 is a flow chart illustrating the creation of a view including dynamic actions.

Figure 7 is a schematic representation of a view record.

Figure 8 is a flow chart illustrating selective show and hide of actions as a user opens a view and selects successive documents within that view.

Best Mode for Carrying Out the Invention

In accordance with a preferred embodiment of the invention, actions which can be performed on a selected document in a view may be dynamically applied to a particular document when that document is selected. Designers can create views and then hide/show actions at the top of the view based on conditions such as values in the document or values derived by formula from some of the fields in the document. Actions may be removed all together, but not just for the entire view, but based on document context. The action list is re-evaluated for each document as documents are selected by arrowing up/down or by clicking, and then the action gets removed if it doesn't apply, or gets added back if it does apply to the selected document.

Herein the term "document" is intended to be generic to anything that may be successively presented in a view content window and with respect to which a plurality of possible actions may be taken. This includes documents, to-do lists, calendar views, databases, spreadsheets, web views, and so forth.

A simple example is a "To Do" view, where some of the tasks are already completed and some are incomplete. It is not needed, and in fact confusing, to display both "mark complete" and "unmark complete" actions at the top of the view. Only one of these actions applies to a given document. In accordance with the present invention, designers can hide the action which does not apply. The same is true in calendar views where the user is either an owner or a participant -- and both of the owner actions and participant actions need not be presented to a specific user. Users are not shown actions that result in "This is not valid for this document", which is annoying. Also, the number of actions presented to the user is reduced, which renders the action bar more logical and easier to interpret.

Referring to Figure 1, a schematic representation of a workstation user interface is shown after the pattern of Lotus Notes. Interface 20 includes control menu icon 22, title bar 24, minimize icon 26, maximize/restore icon 28, close icon 30, work station menu bar 32, smart icon bar 34, page tabs 36, navigator panel 38, view panel 40, preview panel 42, and status bar 44. View panel 40 typically presents a list of documents and several action buttons for selecting various actions to perform on those documents. These component parts of a typical workstation user interface may be present in different combinations, configurations and positions, depending upon the application or database or view being presented, and generally have functions well known to those of skill in the art. While the interface 20 illustrated in Figure 1 is representative of the Lotus Notes user interface, similar components exist in, for example, Microsoft Windows -- to which the present invention is similarly applicable.

Referring to Figures 2 and 3, view panel 40 is presented in accordance with the preferred embodiment of the invention. In Figure 2, document 1 has been selected and the actions displayed as applicable to that document include action 1, action 3, and action 4. Action 2 has been removed for the list, having evaluated as not pertinent. In Figure 3, document 2 has been selected and the actions displayed as applicable to that document are action 2 and action 4. Action 2 has been restored to the set of actions, and action 3 deleted.

Referring to Figure 4, as actions are created by a product designer or application developer for a particular application, in step 92 it is determined if the action is to be static or dynamic. If dynamic, in step 94 the designer/developer creates a hide-when formula.

Referring to Figure 5, an action record 100 is created for each action which includes label 102, text attributes 104 (including fonts, colors, background and border attributes), action buttons 106 which define what is to be done if the action button is selected (such as, reply, forward, create new memo, and so forth), and hide-when formula 108. Formula 108 is set to null for static actions (those which are not to be evaluated dynamically as a user selects successive documents for the view panel 40). Formula 108 for dynamic actions specifies the conditions under which to return true, and typically references fields or data in the relevant document.

Referring to Figure 6, in step 110 an application developer creates a view. Step 114, for each action initialized (Figure 4), determines from action record 100 relevant to this view if the action is static or dynamic. Hide-when formula 108 will be null for static actions. In step 116, for dynamic actions, the developer sets an action to be evaluated flag 128 in view record 120. Once step 116 evaluates to "yes", the process is done -- there is no need to go to a next action.

Referring to Figure 7, view record 120 includes document filter 122, which is a formula in, for example, Lotus Script or Java script, to define a subset of documents -- those documents to be shown in the view; formatting specifications 124; action selection list 126, a list of actions to show in this view; and action to be evaluated flag 128, which returns true or false whether the actions of list 126 are to be evaluated -- this is the evaluate hide-when flag.

Referring to Figure 7, the process executed as a user opens a first 130 and successive documents 132 in a view includes in step 134 evaluating hide-when flag 128 to determine if actions for this view are to be evaluated as dynamic or static. If flag 128 returns no, in step 136 all actions defined for this view are shown. If flag returns yes, in step 138 processing loops through the action records 100 for all actions in selection list 126. If this action has a hide-when formula 108, in step 142 that formula is evaluated. If there is no hide-when formula (field 108 is null) or if the hide-when formula from field 108 returns true, then in step 148, this action is shown in the view. Otherwise, in step 146 the action is hidden, and not presented in the view. There is only one flag in the view which when present forces the view to go through all of the actions for evaluating hide-when.

In most applications, a user is presented a set of actions that can be used. For example, in a to do list, the user is presented with a list of tasks with respect to which several actions may be taken, such as marking incomplete actions completed or other actions incomplete. Previously, the list of available action has been static, and is not reevaluated for each document presented. In accordance with the preferred embodiments of the invention, the list of available actions may be dynamically evaluated for each document brought to view through the use of the hide-when flag for a view and hide-when formula for each action, as discussed above.

Advantages over the Prior Art

It is an advantage of the invention that there is provided an improved system and method for presenting actions which may be executed with respect to a document in a view.

It is an advantage of the invention that there is provided a system and method for dynamically evaluating actions which may be executed with respect to successive documents in a view.

It is an advantage of the invention that there is provided a system and method whereby actions which may be executed with respect to a view may be dynamically evaluated and selectively hidden or shown for successive view panel content.

Alternative Embodiments

It will be appreciated that, although specific embodiments of the invention have been described herein for purposes of illustration, various modifications may be made without departing from the spirit and scope of the invention. In particular, it is within the scope of the invention to provide a computer program product or program element, or a program storage or memory device such as a solid or fluid transmission medium, magnetic or optical wire, tape or disc, or the like, for storing signals readable by a machine, for controlling the operation of a computer according to the method of the invention and/or to structure its components in accordance with the system of the invention.

Further, each step of the method may be executed on any general computer, such as IBM Systems designated as zSeries, iSeries, xSeries, and pSeries, or the like and pursuant to one or more, or a part of one or more, program elements, modules or objects generated from any programming language, such as C++, Java, Pl/1, Fortran or the like. And still further, each said step, or a file or object or the like implementing each said step, may be executed by special purpose hardware or a circuit module designed for that purpose.

Accordingly, the scope of protection of this invention is limited only by the following claims and their equivalents.